

# Construindo Sistemas Multi-Agentes no Contexto de Internet das Coisas

Chrystinne O. Fernandes<sup>1</sup>, Carlos J. P. Lucena<sup>2</sup>

<sup>1</sup>Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) CEP 22453-900 – Rio de Janeiro – RJ – Brasil

{cfernandes, lucena}@inf.puc-rio.br

**Abstract.** *Aiming to expand the set of disciplines that intersects the Multi-Agent Systems (MAS) area, this paper brings the concept of agents to the Internet of Things (IoT). It also presents the results of two case studies, involving a luggage tracking simulation in an airport and a prototype for sensing and monitoring patients and their environment. The solution showed that it is possible to make more proactive environments. The agents were able to detect problems in patients' health status and alert healthcare professionals responsible for their care. The studies assisted modeling and design of a framework for monitoring things based on an agent model.*

**Resumo.** *Visando ampliar o conjunto de disciplinas que tem interseção com a área de Sistemas Multi-Agentes (SMA), este artigo traz o conceito de agentes para a Internet of Things (IoT). Também apresenta os resultados de dois estudos de caso, envolvendo uma simulação de rastreamento de bagagens em um aeroporto e um protótipo para sensoriamento e monitoramento de pacientes e seu ambiente. A solução mostrou que é possível tornar os ambientes mais proativos. Os agentes foram capazes de detectar problemas no estado de saúde dos pacientes e alertar os profissionais responsáveis pelo seu tratamento. O experimento auxiliou na modelagem e projeto de um framework para monitoramento de coisas, baseado em um modelo de agentes.*

## 1. Introdução

Estamos vivendo um período em que a tecnologia está sendo cada vez mais usada para melhorar nossa qualidade de vida, seja aumentando nossa segurança, conforto e bem-estar, seja otimizando recursos humanos limitados como o tempo ou a capacidade de armazenar e processar dados, a fim de extrair deles informação útil e gerar novos conhecimentos. A *Internet of Things* (IoT) é uma área relativamente recente que vem garantindo seu espaço nesse processo de mudanças tecnológicas que se reflete em nossos hábitos cotidianos e em nossa cultura de modo geral.

A ideia básica por trás do conceito de IoT é ter um cenário onde coisas, pessoas e animas são identificados de maneira única e podem comunicar-se pela Internet, com pouca ou nenhuma intervenção humana. Vale salientar que as coisas, no âmbito da IoT, podem ser entendidas como entidades do mundo real que vão fazer parte da rede IoT, compreendendo desde simples objetos como uma cadeira, um eletrodoméstico ou um carro, até seres vivos como plantas, animais e pessoas.

Desse modo, interações na rede, que até então estavam restritas aos tipos *human-to-human* ou *human-to-computer*, estão sendo ampliadas para suportar outros conceitos de interação envolvendo coisas, como os tipos *human-to-thing* e *thing-to-thing*.

Materializando este conceito de Internet das Coisas, atualmente já temos exemplos de sua aplicação em diversas áreas das nossas vidas, como transporte e logística e cuidados médicos. A tendência é que haja uma expansão do alcance da IoT, de modo a possibilitar o surgimento de ambientes cada vez mais inteligentes, os quais vislumbramos que possam evoluir para a dimensão de cidades inteligentes, que é um dos temas de pesquisa amplamente explorados neste campo.

Podemos relacionar diversos problemas que são bastante comuns em nossas cidades e retardam a ocorrência de avanços sociais, como por exemplo: a gestão ineficiente de recursos públicos, que leva a desperdícios de verbas e à consequente insatisfação da população; O alto congestionamento de veículos que impacta negativamente a vida das pessoas, consumindo seu tempo e muitas vezes promovendo stress; O uso de processos manuais de coleta de dados propensos a erros; Além da utilização de técnicas obsoletas de identificação de coisas;

A busca de soluções para problemas desta natureza serve de motivação para este trabalho, cujo enfoque principal consiste no desenvolvimento de sistemas multi-agentes no contexto de Internet das Coisas, a fim de que o conceito de cidades inteligentes torne-se cada vez mais próximo da nossa realidade. Com o intuito de fornecer um suporte ao desenvolvimento de sistemas deste tipo, foi idealizado neste trabalho o projeto de um framework para construção de aplicações IoT, com base em um modelo de agentes de software.

A estratégia utilizada para alcançar este objetivo consistiu, inicialmente, em encontrar cenários com grande potencial para exploração dos conceitos de IoT e que também fossem alvos para a ampla aplicação de agentes. Nesta etapa, foram priorizados dois domínios de aplicação: o primeiro foi mais específico - *track and tracing* -, onde buscou-se analisar o rastreamento de coisas através de simulações; Já o segundo, além de mais geral, foi explorado através de um protótipo construído durante este trabalho para auxiliar no monitoramento de pacientes e correspondeu ao domínio de e-health.

Foram realizados dois estudos de caso, um para cada domínio, onde ambos foram planejados para investigar como se davam as interações *human-to-thing* e *thing-to-thing* em uma rede IoT, além de como os agentes de software iriam se comportar no contexto da IoT e se o uso dos agentes tornaria o ambiente mais proativo, interativo e inteligente. Além do levantamento destas e de outras questões de pesquisa, na fase de planejamento do experimento também foram definidas algumas hipóteses com base no conhecimento prévio sobre o assunto. Tanto as questões quanto as hipóteses serão apresentadas mais adiante, na seção 3. Ao final de cada estudo de caso, foram discutidos os resultados do experimento. Nesta etapa, foram analisadas as hipóteses levantadas, para confirmá-las ou refutá-las, através do seu confronto com os dados obtidos experimentalmente. Esta discussão foi feita com o intuito de tentar responder as questões de pesquisa e registrar possíveis conhecimentos empíricos que possam ter surgido acerca do tema.

A realização dos estudos de caso possibilitou o levantamento de um conjunto de *features* necessárias às etapas iniciais de modelagem e projeto do framework, cuja implementação é um dos trabalhos futuros sugeridos neste artigo.

## **2. Fundamentação Teórica**

### **2.1. Sistemas Multi-Agentes**

Um agente é um elemento de um sistema computacional que está situado em algum ambiente no qual é capaz de realizar ações autônomas, a fim de cumprir os objetivos que

lhes foram delegados. Apesar de não existir uma definição universal do termo agente, existe um consenso na literatura de que a autonomia é uma característica chave para a atuação do agente [Wooldridge 2009]. Neste contexto, autonomia significa a possibilidade de atuação sem a intervenção humana ou de outros sistemas.

Entretanto, mesmo tendo controle sobre seu estado interno e comportamento, o agente não tem controle total sobre seu ambiente. Ele possui um conjunto de ações que pode realizar e cuja execução pode resultar na modificação do ambiente. Por essa razão, considera-se que o agente tem controle parcial sobre seu ambiente, podendo influenciá-lo, de acordo com a ação que decidir realizar. Desse modo, um agente utiliza sua autonomia para decidir como agir de modo a satisfazer seus objetivos. Outras propriedades dos agentes, por sua vez, são desejáveis apenas em domínios específicos, como é o caso do aprendizado, ou seja, a capacidade que possui de aprender a partir da sua experiência.

Um agente pode ocupar diferentes tipos de ambientes, que se classificam de acordo com suas propriedades, em [Russell and Norvig 2013]: **Acessível ou inacessível**, dependendo da informação que o agente possui sobre ele. Diz-se que o ambiente é acessível quando o agente pode obter informações completas, precisas e atualizadas sobre o estado do ambiente; **Determinístico ou não-determinístico**, de acordo com a garantia de efeito que uma determinada ação terá neste ambiente. Um ambiente determinístico é aquele em que qualquer ação tem a garantia de um efeito único – não existe incerteza sobre o estado que irá resultar da execução de uma ação; **Estático ou dinâmico**, dependendo do controle que o agente tem sobre as mudanças no ambiente. Um ambiente pode ser considerado estático quando pode-se assumir que ele irá permanecer inalterado, exceto pela execução de uma ação por parte do agente. Em contrapartida, um ambiente dinâmico é aquele que tem outros processos operando sobre ele e que as mudanças ocorridas nele estão fora do controle do agente; E, por último, um ambiente pode ser **discreto ou contínuo**, considerando as possíveis ações e percepções que o agente pode realizar. Em um ambiente discreto, existe um número finito e fixo de ações e percepções.

Tomando como exemplo os agentes do estudo de caso na área de saúde, podemos considerar seu ambiente inacessível, não-determinístico, dinâmico e contínuo. Neste contexto, é inacessível por não ser possível ter conhecimento completo sobre os dados do ambiente, uma vez que eles refletem o estado de saúde do paciente e são coletados em tempo real. É não-determinístico porque não é possível garantir que cada ação estará associada à exatamente um único efeito possível. Dinâmico, pois o ambiente está em constante mudança, de acordo com eventos que ocorrem e que fogem ao controle dos agentes. E, por fim, contínuo, já que existem infinitas e variáveis ações que podem acontecer neste cenário.

Agentes inteligentes tem capacidades como reatividade, proatividade e habilidade social, as quais diferem pelo modo como os ajudam a satisfazerem seus objetivos. A reatividade diz respeito à capacidade de perceber e responder tempestivamente às mudanças no ambiente. Já a proatividade é a capacidade de ter um comportamento orientado ao seu objetivo, através da tomada de iniciativa. E a habilidade social é a capacidade de interagir com outros agentes e, possivelmente, com humanos. Ela não envolve apenas troca de informações, mas também a possibilidade de realizarem relações de negociação e cooperação [Russell and Norvig 2013].

## 2.2. Inteligência Artificial (IA)

Inteligência Artificial é o campo que tenta não apenas compreender, mas construir entidades inteligentes [Russell and Norvig 2013], que podem ser interpretadas como agentes de software. O conceito de agente é amplamente abordado em IA, bem como o uso de técnicas que podem ser utilizadas pelos agentes para execução de tarefas e tomada de decisões.

Outro aspecto bastante explorado em IA é o processo de aprendizagem. O aprendizado é uma das possíveis características dos agentes. Pode-se dizer que um agente irá aprender se conseguir melhorar o seu desempenho nas tarefas futuras a partir da observação do mundo. Dessa forma, o conhecimento sobre o mundo faz-se necessário para que os agentes possam tomar boas decisões. Este conhecimento é armazenado em uma base de conhecimento, através de uma linguagem de representação do conhecimento. [Russell and Norvig 2013].

## 2.3. Internet of Things

No final da década de noventa, tínhamos o seguinte cenário na Internet: um grande poder computacional estava acessível, mas a entrada de dados na rede era feita quase exclusivamente por seres humanos, que em geral tem recursos de tempo bastante limitados. Este fato gerava um gargalo no canal de entrada de dados na rede, já que a capacidade de processamento era muito superior à demanda de dados. Para tentar resolver limitações como esta, surgiu o conceito de *Internet of Things*, onde coisas também estariam habilitadas a realizar operações de entrada e saída de dados e processamento de informações na Internet.

A ideia era que as coisas fossem identificadas e conectadas à Internet, com a capacidade de se comunicarem de modo independente ou com intervenção mínima dos seres humanos. Esta mudança visava automatizar atividades humanas dispendiosas como a de coletar dados e inseri-los na rede. Desse modo, tarefas como esta passaram a ser delegadas às próprias coisas, que poderiam interagir entre si e também com pessoas e sistemas.

Aplicações IoT tem gerado impacto em diversas áreas da nossa vida: na área pessoal, por exemplo, surgiram os conceitos de *domotics* e *smart house*; No campo profissional, surgiram os *smart offices*; Na Indústria, aplicações no ramo de transporte e logística despontam como grandes promessas para otimização de recursos; Na Administração pública, estamos experimentando aplicações que tratam da gestão mais eficiente de recursos públicos. E, por último, temos a área de saúde, um dos domínios de aplicação dos estudos de caso deste trabalho, que vem explorando conceitos como *e-health* e *Assisted living*.

Existem diversas tecnologias que podem ser utilizadas no desenvolvimento de aplicações IoT. A concepção deste trabalho foi baseada no conceito de três tecnologias que foram fundamentais para embasar nossas simulações: RFID (*Radio-Frequency Identification*), microcontroladores Arduino e Sensores.

RFID ou identificação por rádio frequência é um método de identificação automática que utiliza sinais de rádio, recuperando e armazenando dados remotamente através de dispositivos denominados etiquetas ou tags RFID. Estes dispositivos são usados para fins de identificação, sensoriamento e comunicação [Atzori 2010].

O arduino, por sua vez, é uma plataforma de prototipagem eletrônica, open-source, flexível, disponível na versão stand-alone ou para comunicação com software

executando em computador pessoal. Por serem baseados em hardware e software considerados fáceis de usar, microcontroladores Arduino são destinados a qualquer pessoa que queira desenvolver projetos interativos [Doukas 2012].

A terceira e última tecnologia utilizada foram os sensores. Diversos tipos de sensores podem ser utilizados para coletar dados úteis em aplicações IoT, como a frequência de batimentos cardíacos, nível de açúcar no sangue (para controlar pacientes diabéticos), taxa respiratória, temperatura, umidade, nível de líquido, nível de oxigênio, nível de gás carbônico, acelerômetro, entre outros.

### **3. Metodologia**

#### **3.1. Método Experimental**

Conforme mencionado anteriormente, os métodos utilizados para responder as questões de pesquisa consistiram na realização de dois estudos de caso, abordando os domínios de *track and tracing e e-health*, os quais serão detalhados nas próximas subseções, 3.4 e 3.5.

#### **3.2. Questões de Pesquisa (Q)**

**Q1.** Como os agentes de software comportam-se no contexto da IoT?

**Q2.** O uso dos agentes pode tornar o ambiente mais proativo e interativo?

#### **3.3. Hipóteses (H)**

**H1.** O uso de agentes facilita a comunicação entre os elementos da rede IoT.

**H2.** O uso de agentes pode tornar o ambiente mais proativo.

**H3.** O uso de agentes pode tornar o ambiente mais interativo.

#### **3.4. Estudo de Caso 1: Rastreamento de bagagens**

##### **3.4.1 Definição do Problema**

Atualmente, a grande maioria dos aeroportos ainda não conseguiu superar problemas corriqueiros e que provocam uma série de transtornos aos usuários de seus serviços. Problemas como extravios de bagagens ainda são recorrentes, mesmo com todo o aparato tecnológico que dispomos hoje em dia. Além disso, os passageiros, muitas vezes, veem-se obrigados a acompanharem atentamente os avisos nos monitores espalhados pelo salão de embarque. Isto ocorre porque precisam se certificar de que não houve mudança de última hora em seu portão de embarque, fato que é bastante comum acontecer. Outro inconveniente constante é a poluição sonora provocada pelas últimas chamadas para embarque dos passageiros retardatários e avisos de atraso nos voos.

##### **3.4.2 Solução Proposta**

Como uma alternativa para tentar solucionar problemas como os citados, foi proposto o desenvolvimento de um sistema multi-agentes, no âmbito de Internet das Coisas, através do qual seriam simuladas algumas das situações citadas na subseção anterior. Com isto, pretendia-se, principalmente: Aumentar a inteligência do ambiente simulado, bem como a capacidade de comunicação dos seus elementos (*Smart Environments domain*); Rastrear as bagagens dos passageiros (*Track and Tracing domain*); Monitorar os ambientes nos quais animais eram transportados durante os voos (*Monitoring domain*); Aumentar a confiança dos passageiros no transporte de seus animais, através do sensoriamento de

suas condições físicas e de seu ambiente (*Sensing domain*); Realizar, de forma autônoma, atividade como os alertas aos passageiros feitos pelas companhias aéreas, a partir do uso de agentes de software;

### **3.4.3 Objetivos**

O objetivo deste estudo de caso foi desenvolver um sistema no qual fossem realizadas simulações de situações práticas que ocorrem em aeroportos, com o intuito de alcançar os seguintes resultados específicos: Fornecer estratégias para minimizar problemas recorrentes nos aeroportos; Identificar de forma eficiente os passageiros e suas bagagens; Evitar o extravio de bagagens, assim como a troca de bagagens entre passageiros; Agilizar a retirada das bagagens pelos passageiros na esteira; Permitir o transporte de animais de forma mais segura, com o uso de sensores de temperatura, bem como de oxigênio, de nível de água, entre outros; Diminuir a poluição sonora; Notificar passageiros de forma individual, no que concerne a avisos de atrasos nos voos, abertura de embarques, mudanças nos portões de embarque, últimas chamadas para embarque de passageiros retardatários; E, por último, realizar os créditos correspondentes aos voos dos passageiros, em seus planos de milhas, de forma automática, no momento do check-in;

## **3.5 Estudo de Caso 2: Monitoramento de pacientes**

### **3.5.1 Definição do Problema**

Muitos são os desafios evidenciados pela área de saúde, visto que se trata de um ambiente onde os profissionais lidam com uma questão delicada que é a vida dos pacientes e com os riscos envolvidos em seus tratamentos médicos, desde exames de rotina a procedimentos cirúrgicos de mais alta gravidade. Em consequência de razões como esta, os avanços tecnológicos acabam ocorrendo em um ritmo mais lento que em outras áreas, já que se exige mais cautela ao lidar com o estado de saúde dos pacientes, que muitas vezes já encontram-se debilitados, física e emocionalmente.

Além disso, os entraves ao desbravamento deste meio não se limitam ao campo de atuação da atividade estritamente médica. Entram em cena questões éticas que precisam ser resolvidas antes mesmo de dar início a qualquer método de pesquisa científica que tenha participação direta dos pacientes. Nas universidades, tem-se os comitês de ética para avaliar se o trabalho respeita as regras preconizadas pelos regulamentos de ética. Desse modo, o aluno precisa submeter seu trabalho a este comitê para conseguir aprovação para avançar em suas pesquisas. Uma vez autorizado pelo comitê da academia, o pesquisador irá precisar também da autorização do responsável pelo hospital ou unidade de tratamento hospitalar em que ocorrerá a pesquisa. Todas estas questões devem ser consideradas na etapa de planejamento do método de pesquisa, para que seja viabilizada a aplicação de seu experimento no ambiente real.

De modo mais específico, a motivação para realizar este estudo de caso veio de problemas como a predominância da reatividade, em detrimento da proatividade, no ambiente em que o paciente está sendo tratado. O que significa, por exemplo, que, em muitos casos, medidas são tomadas pela equipe médica somente depois que o paciente já teve o seu quadro piorado de alguma forma.

Fatos como este são bastante problemáticos, principalmente porque, muitas vezes, em virtude do tempo decorrido entre o momento em que o paciente teve complicações em seu estado de saúde e o momento em que este fato foi detectado pelos profissionais de saúde responsáveis por agir nestas circunstâncias, que aqui estamos chamando de

**Detection Anomaly Interval (DAI)**, é grande o suficiente para que se tenham consequências indesejáveis, podendo provocar sequelas nos pacientes ou até mesmo levá-lo à óbito, no pior dos casos.

Um dos fatores que contribuem para que o DAI tenha uma forte tendência em apresentar um alto valor é o fato de que a inspeção do estado de saúde dos pacientes ainda é comumente realizada de modo presencial. Desta forma, o profissional da saúde precisa se deslocar até o local em que se encontra o paciente e inspecioná-lo, coletando seus dados, através da medição de seu pulso, temperatura, dentre outros dados vitais.

### **3.5.2 Solução Proposta**

A solução planejada neste experimento consiste no levantamento de alternativas para tratar de problemas da ordem dos que foram mencionados anteriormente, como o grande lapso de tempo que se verifica no processo de detecção de anomalias. Neste contexto, anomalias devem ser interpretadas como qualquer anormalidade perceptível no estado de saúde do paciente, da qual decorra a necessidade de intervenção pelo profissional da saúde responsável pelos seus cuidados, no instante em que elas ocorrem.

### **3.5.3 Objetivos**

De modo mais geral, deseja-se tornar o ambiente de atendimento hospitalar mais proativo. Também pretende-se aumentar a capacidade de interação entre as coisas do ambiente, que serão apresentadas na subseção 4.2.1. Com relação aos objetivos específicos, temos, por exemplo, a pretensão de tentar diminuir o intervalo de detecção de anomalias através da atuação dos agentes de software.

## **4. Resultados**

### **4.1 AirportTrack&Trace System**

O primeiro estudo de caso resultou no desenvolvimento de um sistema multi-agentes que consistiu em uma simulação das interações das coisas no aeroporto. Para isto, foram modelados e implementados agentes de software reativos, os quais serão relacionados a seguir.

#### **4.1.1 As Coisas no AirportTrack&Trace System**

Foram modeladas as seguintes coisas no AirportTrack&Trace System: 1- Passageiros com cartões de embarque eletrônicos; 2- Bagagens com identificadores; 3- Animais com identificadores; 4- Voos; 5- Portões de embarque; 6- Esteiras rolantes.

#### **4.1.2 Os Agentes no AirportTrack&Trace System**

Foram modelados os seguintes agentes de software no sistema:

**Tabela 1: Agentes modelados no AirportTrack&Trace System com seu tipo de comportamento baseado no modelo de comportamento do JADE**

Identificador do Agente	Descrição	Behaviour
<i>AnimalEnvironmentReportAgent</i>	Agente de transmissão de informações de sensoriamento do ambiente onde animais são transportados nos voos	<i>TickerBehaviour</i>
<i>UpdateGateAgent</i>	Agente de notificação de mudança no portão de embarque	<i>TickerBehaviour</i>
<i>CallForPassengerAgent</i>	Agente de chamadas de embarque para passageiros retardatários	<i>TickerBehaviour</i>
<i>CreditMilesAgent</i>	Agente de crédito automático de milhas no programa de milhas dos passageiros após check-in	<i>OneShotBehaviour</i>
<i>BaggageClaimAgent</i>	Agente de notificação de proprietários de bagagens	<i>OneShotBehaviour</i>
<i>NotifyBeginBoardingAgent</i>	Agente de notificação de início de embarque	<i>TickerBehaviour</i>
<i>NotifyDelayInFlightAgent</i>	Agente de notificação de atrasos no voo	<i>OneShotBehaviour</i>
<i>BaggageAgent</i>	Agente de alerta de desvio de rota da bagagem	<i>TickerBehaviour</i>
<i>AnimalEnvironmentSensorAgent</i>	Agente de sensoriamento do ambiente onde estão sendo transportados os animais	<i>TickerBehaviour</i>

Para construção do AirportTrack&Trace System, foi utilizada a linguagem de programação Java. A arquitetura do sistema foi projetada de modo que este fosse composto por módulos com responsabilidades bem definidas. Os agentes foram modelados no sistema a fim de compor um módulo específico, que conteve apenas agentes reativos e foi implementado utilizando-se a versão 4.3.0 da ferramenta JADE (*Java Agent DEvelopment Framework*). Jade é um framework para desenvolvimento de aplicações de agentes em Java, que simplifica a implementação de sistemas multi-agentes através de um middleware que está em conformidade com o padrão FIPA (*Foundation For Intelligent, Physical Agents*) e utiliza um conjunto de ferramentas gráficas que suportam as fases de depuração e implantação.

A API JADE disponibiliza dois tipos de classes de comportamentos que podem ser estendidos para os agentes: Primitivos e Compostos. O comportamento dos agentes do AirportTrack&Trace System foi implementando a partir deste modelo de comportamentos, mais especificamente da classe de comportamentos primitivos (*Primitive Behaviours*). Cada agente do AirportTrack&Trace System é uma extensão da classe *Agent* e possui um comportamento que corresponde à extensão da classe *Behaviour* do JADE. Todo agente do sistema tem seu comportamento definido através do seu método *setup*, onde seu comportamento é configurado, através do método *addBehaviour*.

O AirportTrack&Trace System possui agentes reativos com dois tipos de comportamentos: *TickerBehaviour* e *OneShotBehaviour*. Como comportamentos do tipo *TickerBehaviour* executam suas ações de forma cíclica, escolhemos os agentes do nosso cenário que precisam realizar atividades de monitoramento contínuo para implementar este comportamento, como é o caso dos agentes: 1- *CallForPassengerAgent*: Agentes que precisam monitorar continuamente se todos os passageiros que fizeram check-in em determinado voo já embarcaram, a fim de notificar retardatários; 2- *NotifyBeginBoardingAgent*: Agentes que checam, de tempos em tempos, o início de embarques; 3- *UpdateGateAgent*: Agentes que verificam constantemente a ocorrência de mudanças no portal de embarque;

Outros agentes, por sua vez, são encarregados de executar tarefas que não são realizadas em um intervalo de tempo predefinido, ou seja, que ocorrem sob demanda, em

resposta a um determinado evento. É o caso do agente *CreditMilesAgent*, que credita as milhas no plano de milhas do passageiro no momento em que este realiza o check-in;

#### 4.1.3 Discussão

As simulações realizadas no sistema mostraram que muitos dos problemas apresentados neste cenário poderiam ser minimizados adotando-se estratégias simples, como o uso de um sistema que abrangesse funcionalidades como as do AirportTrack&Trace System: Identificação de bagagens dos passageiros com etiquetas RFID, para serem monitoradas e rastreadas a fim de evitar extravios; Uso de identificação por radiofrequência também para os animais que estão sendo transportados, para que informações sobre as suas condições físicas e de seu ambiente possam ser coletadas, transmitidas e monitoradas para aumentar a segurança dos animais durante os voos e, conseqüentemente, a confiança dos seus donos; Criação de um cartão de embarque eletrônico para passageiros (*BoardingPass*), para facilitar o processo de comunicação das empresas aéreas com os passageiros dos seus voos. Esta troca de informações em um canal direto passageiro-companhia aérea poderia diminuir a poluição sonora decorrente das chamadas para embarque de passageiros retardatários e demais avisos. Também evitaria que os passageiros verificassem continuamente a ocorrência de alterações em seus horários de voos, mudanças em seus portões de embarque e outras notificações importantes.

## 4.2 MonitoringPatient System

### 4.2.1 As Coisas no MonitoringPatient System

Os seguintes elementos foram modelados no sistema para representar o conceito das coisas envolvidas na unidade de tratamento hospitalar onde o paciente irá receber o atendimento médico: Pacientes; Profissionais de saúde; Ambiente do paciente (UTI, CTI, Sala de cirurgia, apartamento). Identificadas no sistema de forma única, as coisas serão capazes de comunicarem-se entre si para aumentar a inteligência do ambiente.

### 4.2.2 Os Agentes no MonitoringPatient System

Foram modelados os seguintes agentes de software no sistema:

**Tabela 2: Agentes modelados no MonitoringPatient System com seu tipo de comportamento baseado no modelo de comportamentos do JADE**

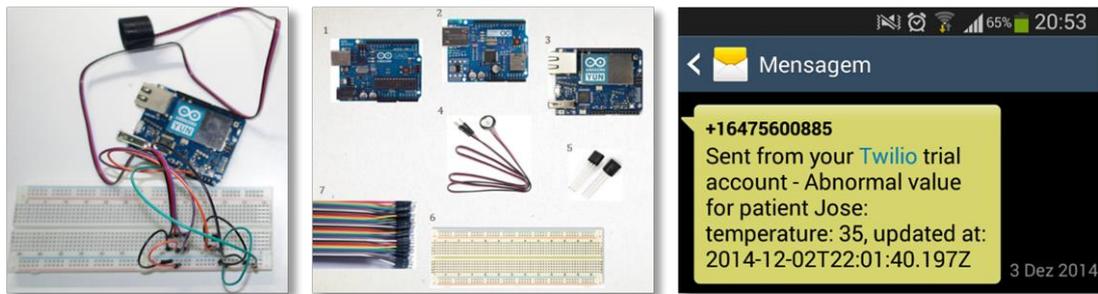
Identificador do Agente	Descrição	Behaviour
VerifySensorDataAgent	Procura anomalias nos dados sensoreados	<i>TickerBehaviour</i>
SendSMSAgent	Notifica anomalias, através do envio de mensagens	<i>OneShotBehaviour</i>

A construção do sistema MonitoringPatient System também foi feita a partir da linguagem Java. O sistema também dispunha de um módulo de agentes que foi implementado utilizando o framework JADE. Este módulo é composto pelos agentes reativos: VerifySensorDataAgent e SendSMSAgent.

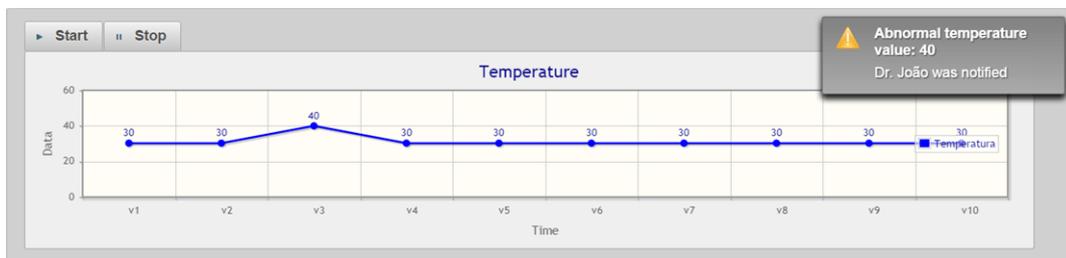
O agente VerifySensorDataAgent, conforme seu identificador sugere, possui a tarefa de monitorar os dados vitais que estão sendo sensoreados. Para que esta solução funcione, é preciso que se utilize uma tecnologia muito comum no âmbito da IoT, que são os microcontroladores. A ideia é que sejam adicionados ao microcontrolador, sensores de diferentes tipos, como sensor de batimento cardíaco e temperatura, por exemplo, formando, assim, um dispositivo IoT capaz de coletar os dados paciente automaticamente (Figura 1-a). Uma vez coletados, os dados são armazenados em uma

base de dados, de modo que o agente *VerifySensorDataAgent* tenha acesso e consiga monitorá-los para detectar possíveis anomalias. A figura 2 mostra um exemplo de atuação deste agente, onde ele detecta um valor anormal na temperatura de um dado paciente.

Uma vez detectada uma anomalia, o agente *VerifySensorDataAgent* comunica-se com o agente *SendSMSAgent*, enviando uma mensagem que informa sobre a necessidade de alertar o profissional da saúde responsável por agir no tratamento da anomalia detectada. Deste modo, o agente *SendSMSAgent* envia o alerta para o responsável, através de e-mail ou SMS. A Figura 1-c mostra um exemplo de mensagem SMS enviada pelo agente *SendSMSAgent* ao detectar anormalidade na temperatura de um dado paciente.



**Figura 1-a.** Protótipo IoT para o monitoramento de pacientes. **Figura 1-b.** Elementos do protótipo: 1-Arduino Uno R2; 2- Shield Ethernet compatível com Arduino; 3-Arduino Yún; 4-Sensor de pulso; 5-Sensores analógicos de temperatura; 6-Protoboard; 7-Fios do tipo macho-macho; **Figura 1-c.** Alerta enviado ao profissional de saúde pelo agente *SendSMSAgent*, via SMS.



**Figura 2.** Gráfico usado para plotar os dados de temperatura do paciente. Pode ser acessado a partir do módulo de visualização do *MonitoringPatient System*. No canto superior direito, detecção de anomalia na temperatura do paciente pelo agente *VerifySensorDataAgent*.

### 4.2.3 Discussão

Através das simulações realizadas no sistema, pudemos observar que o DAI poderia ser diminuído com o uso do *MonitoringPatient System*. Esta diminuição traz como consequência o aumento da característica de proatividade do ambiente.

## 4.3 Framework

### 4.3.1 Features (F)

Após a realização dos estudos de caso, foi projetada a primeira versão do framework para construção de sistemas multi-agentes no âmbito da IoT. Nesta versão inicial, modelamos as seguintes *features*, para o núcleo das aplicações:

**F1.** Identificação automática das coisas do domínio através de tags RFID;

- F2. Coleta de dados automática através do uso de protótipos de hardware compostos por microcontroladores Arduino e sensores de diversos tipos e com finalidades variadas;
- F3. Armazenamento e recuperação dos dados sensoreados na base de dados;
- F4. Visualização dos dados da base de dados no sistema, sob a forma de *line charts*;
- F5. Inserção de conhecimento do especialista no domínio na base de dados;
- F6. Definição de estratégias de negociação por recursos para os agentes cognitivos;

### 4.3.2 Arquitetura dos Sistemas Multi-agentes

O framework foi projetado de modo que a arquitetura dos sistemas construídos a partir dele tenha uma estrutura modular. Basicamente, o núcleo principal de cada SMA será composto por três módulos principais:

**Módulo 1. Coleta de Dados:** responsável por receber os dados coletados pelo protótipo, formado pelo microcontrolador e conjunto de sensores específicos da aplicação, e armazená-los na base de dados.

**Módulo 2. Visualização:** corresponde à parte do sistema multi-agentes que irá recuperar os dados sensoreados na base de dados e exibi-los, via interface gráfica. Os dados serão exibidos na forma de gráficos do tipo *line chart*, conforme ilustrado na figura 2.

**Módulo 3. Atuação dos Agentes:** irá tratar das tarefas específicas dos agentes. Pode compreender a atuação de agentes reativos e cognitivos.

No caso de agentes reativos, seu comportamento deverá ser inteiramente mapeado na aplicação, previamente. Desta forma, a reação do agente deve ser definida previamente para possibilitar a sua atuação. Tomando como exemplo o sistema MonitoringPatient, à cada anomalia detectada, precisaria ser definida na configuração inicial da aplicação, o comportamento que deveria ter o agente. A realização desta etapa de configuração precisa ser feita com o auxílio de um especialista no domínio. Os agentes cognitivos, por sua vez, poderão atuar em processos de negociação por recursos.



Figura 3. Visão geral da arquitetura dos SMAs construídos a partir do framework.

## 5 Conclusões e trabalhos futuros

Os estudos de caso mostraram que o DAI poderia ser diminuído com o uso do MonitoringPatient System, aumentando a proatividade do ambiente. Também concluiu-se que aumentou a interação *human-to-thing*, visto que os agentes, considerados coisas do domínio, foram capazes de alertar os profissionais da saúde sobre anomalias no estado de saúde dos pacientes. Além do aumento na interação *thing-to-thing*, já que os agentes

*VerifySensorDataAgent* e *SendSMSAgent*, ambos considerados como coisas do domínio, comunicaram-se efetivamente na realização de suas ações.

Consideram-se, como trabalhos futuros: implementação do framework projetado e aprofundamento nas investigações através do uso de técnicas de *data mining* nos dados coletados para extração de conhecimento útil, bem como de *machine learning*, para que os agentes tenham capacidade de fazer previsões sobre o estado de saúde dos pacientes;

Também pretende-se implementar o comportamento de agentes cognitivos para atuarem na negociação por recursos. No caso de sistemas como o MonitoringPatient System, o agente poderia ser responsável por adotar estratégias de negociação para conseguir recursos hospitalares para um dado paciente. Esta funcionalidade poderia ser modelada de modo que, à cada paciente fosse associado um agente que seria responsável por representá-lo no processo de negociação. Este agente teria acesso às informações referentes aos sinais vitais que estão sendo coletados e, com isso, poderia usar estes dados para gerar conhecimento sobre o estado de saúde do paciente. De posse desse conhecimento, o agente seria capaz de atuar em benefício do paciente que ele representa e requerer acesso a um dado recurso, à medida que este se fizesse necessário.

## Referências

- Atzori, Luigi; Iera, Antonio; Morabito, Giacomo. (2010) “The Internet of Things: A survey”, In: Journal Computer Networks: The International Journal of Computer and Telecommunications Networking, vol.54, pages 2787-2805
- Bui, Nicla; Zorzi, Michele. (2011). “Health Care Applications: A Solution Based on The Internet of Things”. In: ISABEL ‘11 Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologie.
- Weiser, Mark. (1993). “Some Computer Science Issues in Ubiquitous Computing”.
- Kuniavsky, Mike. (2010). “Smart Things: Ubiquitous Computing User Experience Design”.
- Russell, Stuart; Norvig, Peter. (2013). “Inteligência Artificial”. Elsevier.
- Doukas, Charalampos. (2012). “Building Internet of Things with the Arduino”.
- Wooldridge, Michael. (2009). “An Introduction to MultiAgent Systems”. Wiley.
- Padgham, Lin; Winikoff, Michael (2004). “Developing Intelligent Agent Systems”. Wiley
- McRoberts, Michael. (2011). “Arduino Básico”. Novatec.
- Finkenzeller, Klaus. (2003). “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification”. John Wiley & Sons, Inc., NY, USA, 2<sup>nd</sup> edition.
- Nguyen, Stéphanie. (2009). “RFID for Track & Trace of Baggage in Airports. Business Aspects of the Internet of Things, Seminar of Advanced Topics”, In: ETH Zurich, FS2009, Florian Michahelles.
- Zhang, Ting; Ouyang, Yuanxin; He, Yang. (2008). “Traceable Air Baggage Handling System Based on RFID Tags in the Airport”. In: Journal of Theoretical and Applied
- <http://arduino.cc/>
- <http://jade.tilab.com/>